New Signature Derivation using Existing Signatures

N.R.Sunitha [1] and B.B.Amberker [2]
[1] Siddaganga Institute of Technology, Department of Computer Science & Engg., Tumkur, Karnataka, India.
Email: nrsunitha@gmail.com
[2] National Institute of Technology, Department of Computer Science & Engg., Warangal, Andhra Pradesh, India.
Email: bba@nitw.ac.in

***Abstract*— In banks, as part of normal procedure, receipts for deposits, statements of the bank account or credit card account are regularly issued to customers. This whole procedure is time consuming. Also, officials often find it difficult to sign for all the documents required by a customer though the related sub-processes are completed and corresponding documents are digitally signed. We consider the scenario of e-receipt generation during e-cheque processing, where the subprocess like e-cheque verification and receiving acknowledgement from cheque clearing bank are completed and digitally signed. But there is need for e-receipt to be generated by the bank for the customer. When the number of e-cheques increase, it is a burden for the bank to issue e-receipts. In this scenarios, we observe that, it would be interesting if customers themselves are capable of generating signed receipts based on the signatures available on already completed transactions. This calls for signature of a document to be derived from existing signatures of related documents. By this a customer can derive signatures on his own without the intervention of the bank which inturn reduces the work load on the bank. In all the signature derivations we make, we take care that a new signature derived is similar to the one that the signer would have generated if he had signed himself and also all signatures either existing or derived are verified using the same verification equation.**

***Index Terms*— e-banking, e-cheque, Digital Signature, Signature derivation, public key**

I. Introduction

In banks, as part of normal procedure, receipts for deposits, statements of the bank account or credit card account are regularly issued to customers [6, 7]. This whole procedure is time consuming and paper intensive. It would be interesting if customers themselves are capable of generating such signed receipts and bank statements based on the signatures available on already completed transactions. This calls for methods to derive new signatures from existing signatures.

The first part of our paper discusses on deriving a new signature from two existing signatures. Here the first signature is obtained on message m1. The second signature is obtained on message m2. Supposing a signature is required on m1,m2, the signer will generate signature as he had generated the first and second signatures using his secret key. We propose a method by which anyone can derive the signature on m1,m2 using the first and second signatures without the signer intervention.

We extent the method to derive one signature from n existing signatures. We apply this method to automatically generate receipts by payees of cheques after depositing the cheque. The motivation for this idea is derived from [1], where the authors derive a new signature from existing signatures using the property of transitive closure of a graph.

Before arriving at these methods of signature derivation, we initially used basic signature schemes like ElGamal and DSA [8,5,4] signature schemes for signing the messages and later tried to derive new signature from existing signatures. Though a new signature was derived and verification equation obtained, but the problem was, we were unable to derive a new signature similar to the one that the signer would have generated if he had signed himself. Also, the verification equation was different for signer signed messages and derived signatures. In the following sections, in all the signature derivations we consider, we take care that a new signature derived is similar to the one that the signer would have generated if he had signed himself and also all signatures either existing or derived are verified using the same verification equation.

The organisation of our paper is as follows: In Section II, we discuss a method to derive a new signature from *n* existing signatures and apply the concept of deriving signatures on e-receipts for e-cheques submitted to banks. In Section III, we extend the same method to continuously derive new signatures from existing and derived signatures. Lastly, we conclude.

II SIGNATURE DERIVATION ON E-RECEIPTS FOR E-CHEQUES SUBMITTED TO BANKS

When somebody gives us a cheque, we see that it is deposited in our bank so that the cheque gets cleared from the payer's bank and the cheque amount is deposited in our account. During this process, when we submit the cheque we expect a signed receipt to be issued by the bank. When the number of cheques submitted increases, it is a burden for the bank to issue these receipts. To address this problem we process the cheques electronically (e-cheques) and generate e-receipts. We expect the e-receipt to contain the e-cheque details, a message stating that the e-cheque is verified, e-cheque details sent to clearing bank and an acknowledgement from the clearing bank, all digitally signed by the

DOI: 01.ijsip.01.01.02

servicing bank. We propose to use the property of signature derivation to generate e-receipts.

In e-cheque processing, the payee of e-cheque submits the e-cheque details (let us call this m1) to his servicing bank. The servicing bank verifies the cheque details and signs the message "Cheque verified" (let us call this m1'). Later the bank sends the relevant e-cheque details (let us call this m2) to the cheque clearing bank which inturn sends a signed acknowledgement message for receiving the e-cheque. As a customer trusts his own servicing bank than the cheque clearing bank, there is need for the servicing bank to sign the acknowledgement message (let us call this m2') for the e-cheque details sent. During cheque processing, though the messages m1,m1', and m2,m2' are already separately signed, for a receipt to be generated, there is need for a single signature on all the messages i.e. m1,m1',m2,m2'. By having single signature the space to store the signature is reduced and also later for verification of the receipt, a single verification will be sufficient. We propose to derive a single signature on m1,m1',m2,m2' using the existing signatures on m1,m1', and m2,m2'.

In this section, we propose a method to derive a new signature from existing n signatures. The derived signature is on all the messages of the existing signatures. We do not perform any operation like concatenation or addition on the messages. By deriving a new signature, we only reduce the number of signatures.

A. Signing algorithm for n pairs of messages:

We use the idea of generating secret keys and public key from [1, 2]. To sign a pair of messages ($m_1,m_{1'}$), where m_1 can be considered as sender's data and $m_{1'}$ as signer's data, we need to have two pairs of private keys (x_i, y_i), (x_j , y_j) by choosing independently at random from Z_q. Their corresponding public keys v_i, v_j are computed as

$$vi = g^{xi}.h^{yi}$$

$$vj = g^{xj} .h^{yj}$$

where *g* and *h* are the generators of the subgroup G_q of order q of Zp^* . The signature on ($m_1,m_{1'}$), is given by ($\alpha_{i,j}$, $\beta_{i,j}$, $\gamma_{i,j}$,$m_1,m_{1'}$), where

$$\alpha i,j = H(m1) + (xi - xj)$$

$$\beta i,j = H(m1') + (yi - yj)$$

$$\gamma i,j = g^{H(m1)}.h^{H(m1')}$$

where *H(m)* is a hash function [3].

To sign another pair of messages ($m_2,m_{2'}$), where m_2 can be considered as sender's data and $m_{2'}$ as signer's data, we can utilize one of the pairs of previously used private keys say (x_j , y_j) and generate another pair (x_k, y_k) as earlier. The corresponding public key v_k is computed as

$$vk = g^{xk} .h^{yk}$$

.

The signature on ($m_2,m_{2'}$) is given by ($\alpha_{j,k}$, $\beta_{j,k}$, $\gamma_{j,k}$,$m2,m_{2'}$), where

$$\alpha j,k = H(m2) + (xj - xk)$$

$$\beta j,k = H(m2') + (yj - yk)$$

$$\gamma j,k = g^{H(m2)}.h^{H(m2')}$$

In this way any number of pairs of messages can be signed.

The signature on n pairs of messages (m1,m1', . . . ,mn,mn') with the individual pairs of messages already signed by the signer, with n + 1 pair of secret keys (x0, y0), . . . , (xn, yn) and n + 1 public keys v0, . . . , vn is given by (α0,n, β0,n, γ0,n, (m1,m1', . . . ,mn,mn')) where,

$$\alpha 0,n = H(m1) + . . . + H(mn) + (x0 - xn)$$

$$\beta 0,n = H(m1') + . . . + H(mn') + (y0 - yn)$$

$$\gamma 0,n = g^{H(m1)+...+H(mn)} .h^{H(m1')+...+H(mn')}$$

B. Signature derivation for n pairs of messages

We first discuss how to derive a signature using two existing signatures. Let ($\alpha_{i,j}$, $\beta_{i,j}$, $\gamma_{i,j}$,$m_1,m_{1'}$) be the first signatures and ($\alpha_{j,k}$, $\beta_{j,k}$, $\gamma_{j,k}$,m2,$m_{2'}$) be the second signature. The derived signature will be of the form (i, k, $\alpha_{i,k}$, $\beta_{i,k}$, $\gamma_{i,k}$,$m_1,m_2,m_{1'},m_{2'}$) where,

$$\alpha i,k = \alpha i,j + \alpha j,k$$

$$= H(m1) + (xi - xj) + H(m2) + (xj - xk)$$

$$= H(m1) + H(m2) + (xi - xk)$$

$$\beta i,k = \beta i,j + \beta j,k$$

$$= H(m1') + (yi - yj) + H(m2') + (yj - yk)$$

$$= H(m1') + H(m2') + (yi - yk)$$

$$\gamma i,k = \gamma\ i,j.\ \gamma\ j,k$$

$$= g^{H(m1)} .h\ H^{(m1')} .g^{H(m2)} .\ h^{H(m2')}$$

$$= g^{H(m1)+H(m2)} .\ h^{H(m1')+H(m2')}$$

If the signer himself signs for the message (m1, m2, m1',m2'), then he generates the signature (i, k, αi,k, β i,k, γi,k,m1,m2,m1',m2') where

$$\alpha i,k = H(m1) + H(m2) + (xi - xk)$$

$$\beta i,k = H(m1') + H(m2') + (yi - yk)$$

$$\gamma i,k = g^{H(m1)+H(m2}) .\ h^{H(m1')+H(m2')}$$

DOI: 01.ijsip.01.01.02

We observe that the derived signature is identical to the signature generated by the signer.

To derive a single signature (α0,n, β0,n, γ0,n, (m1,m1’, . . . ,mn,mn’)) using *n* existing signatures of the above form, we have

$$\alpha 0,n = \alpha 0,1 + \ldots + \alpha n-1,n$$

$$\beta 0,n = \beta 0,1 + \ldots + \beta n-1,n$$

$$\gamma 0,n = \gamma 0,1 . \ldots . \gamma n-1,n$$

C. Verification of either existing or derived signature

The general equation to verify any signature (i, j, α i,j , β i,j , γ i,j ,m1,m1’) which could be either an existing or a derived equation is as follows,

$$vi.\ \gamma\ i,j = vj.\ g^{\alpha\ i,j} . h^{\beta\ i,j} \qquad (1)$$

$$RHS = g^{xj} .h^{yj} .g^{H(m1)+(xi-xj}) .h^{H(m1’)+(yi-yj\ )}$$

$$= g^{xj} .h^{yj} .g^{H(m1)} .g^{xi} .g^{-xj} .h^{H(m1’).}h^{yi}.h^{-yj}$$

$$= g^{H(m1)} .h^{H(m1’)} .g^{xi}.h^{yi}$$

$$= \gamma\ i,j.vi$$

$$= LHS.$$

D. e-Receipt generation

When a payee submits a cheque, the bank creates the first pair of secret keys (xi, yi) by choosing independently at random from Zq. g and h are the generators of the subgroup Gq of order q of Zp* . The public key vi is computed as vi = g^{xi} . h^{yi} . The cheque details are available in m1. The bank creates the second pair of secret keys (xj , yj) as earlier and computes the public key vj as vj = gxj .hyj . The bank verifies the cheque and generates a message m1’ which contains the message saying that the cheque is verified. It creates the first signature on messages m1 and m1’ using the first and second secret key pairs. The signature is (i, j, α i,j , βi,j , γ i,j ,m1,m1’), where

$$\alpha\ i,j = H(m1) + (xi - xj)$$

$$\beta i,j = H(m1’) + (yi - yj)$$

$$\gamma\ i,j = g^{H(m1)} .h^{H(m1’)}$$

The bank publishes this signature for the payee of the cheque. Now the bank submits the cheque to the payer’s bank and gets the acknowledgement message for cheque submission. The bank creates the third pair of secret keys (xk, yk) as earlier and computes the public key vk as vk = g^{xk} .h^{yk} . The bank creates a second signature (α j,k, βj,k, γj,k, m2, m2’) using the second and third secret key pairs where m2 indicates the cheque details sent by bank to payer’s bank and m2’ indicates the acknowledgement message received from payer’s bank for cheque submission. The other components of the signature are computed as follows:

$$\alpha\ j,k = H(m2) + (xj - xk)$$

$$\beta j,k = H(m2’) + (yj - yk)$$

$$\gamma\ j,k = g^{H(m2)} .h^{H(m2’)}$$

This signature is also published by the bank.

Generally the bank is expected to issue a receipt to the payee for cheque submission. In case the bank issues a receipt with signature on m1,m1’,m2,m2’, the signature can be generated using the first and the third secret key pairs as follows:

(i, k, α i,k, β i,k, γ i,k,m1,m2,m1’,m2’) where

$$\alpha\ i,k = H(m1) + H(m2) + (xi - xk)$$

$$\beta\ i,k = H(m1’) + H(m2’) + (yi - yk)$$

$$\gamma\ \ i,k = g^{H(m1)+H(m2)} .\ h^{H(m1’)+H(m2’)}$$

But as the number of cheque submissions increase, it becomes tedious to issue receipts for all payees of cheques. Therefore we propose to derive the above signature using the first and second signatures published by the bank using the following equations:

$$\alpha\ i,k = \alpha\ i,j + \alpha\ j,k$$

$$= H(m1) + (xi - xj) + H(m2) + (xj - xk)$$

$$= H(m1) + H(m2) + (xi - xk)$$

$$\beta i,k = \beta i,j + \beta j,k$$

$$= H(m1’) + (yi - yj) + H(m2’) + (yj - yk)$$

$$= H(m1’) + H(m2’) + (yi - yk)$$

$$\gamma\ i,k = \gamma\ i,j.\ \gamma\ j,k$$

$$= g^{H(m1)} .\ h\ H(m1’)\ \ .\ g\ H(m2).h\ H(m2’)$$

$$= g^{H(m1)+H(m2)} .\ h^{H(m1’)+H(m2’)}$$

Any signature can be verified using equation (1).

As the first and second signatures related to the cheque processing are already done and published by the bank when the relevant process is completed, the payee of the cheque can generate the receipt on his own without the bank’s intervention. Thus the load on the bank to generate receipts is totally removed.

III CONTINUOUS DERIVATION OF NEW SIGNATURES FROM EXISTING AND DERIVED SIGNATURES

Here we extend the method discussed in the previous section to derive new signature from derived and existing signatures. This helps us to continuously derive signatures on the new messages generated. Alice creates

DOI: 01.ijsip.01.01.02

an initial node i with secret keys (xi, yi) (see Figure1), where (xi, yi) is chosen independently at random from Zq. g and h are the generators of the subgroup Gq of order q of Zp^* . The public key vi is computed as vi = $g^{xi}.h^{yi}$. To process a transaction T1 of customer, Alice creates a node j with secret keys (xj , yj) and public key vj computed as vj = $g^{xj}.h^{yj}$. Let m1 be the data message sent by the customer and m1’ be the numerical value related to transaction T1. To sign the messages m1,m1’, Alice creates the signature (i, j, α i,j , βi,j , γ i,j ,m1,m1’), where

$$\alpha\ i,j = H(m1) + (xi - xj)$$

$$\beta i,j = H(m10) + (yi - yj)$$

$$\gamma\ i,j = m1' g^{H(m1)} . h^{H(m1')}$$

When compared to the previous method of generating new signature, we have modified the equation of γ i,j by multiplying with m1’, which later helps the customer to substitute the data message received from the signer in the verification of signature and verify its validity. This signature is published by Alice.

For the second transaction T2 Alice creates another node k with secret keys (xk, yk) and public key vk = $g^{xk}.h^{yk}$. Let m2 be the data message sent by the customer and m2’ be numerical value related to transaction T1. To sign the messages m2,m2’, Alice creates the signature (j, k, α j,k, β j,k, γj,k,m2,m2), where

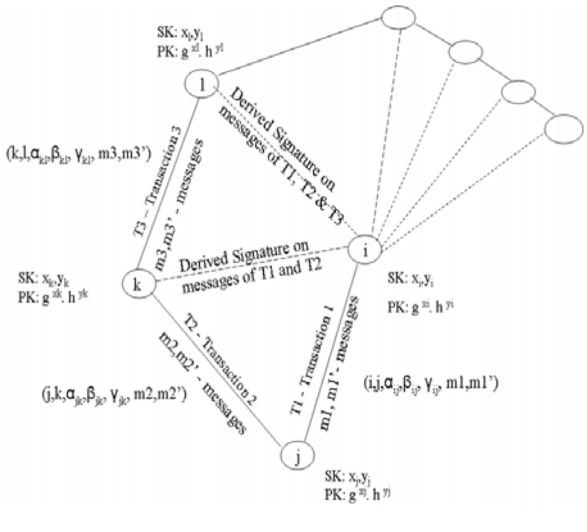
SK: x_l,y_l
PK: $g^{xl}.h^{yl}$
l
Derived Signature on messages of T1, T2 & T3
(k,l,α_{kl},β_{kl}, γ_{kl}, m3,m3’)
T3 – Transaction 3
m3,m3’ - messages
Derived Signature on messages of T1 and T2
i
SK: x_k,y_k
PK: $g^{xk}.h^{yk}$
k
SK: x_i,y_i
PK: $g^{xi}.h^{yi}$
T2 - Transaction 2
m2,m2’ - messages
T1 - Transaction 1
m1, m1’- messages
(i,j,α_{ij},β_{ij}, γ_{ij}, m1,m1’)
(j,k,α_{jk},β_{jk}, γ_{jk}, m2,m2’)
j
SK: x_j,y_j
PK: $g^{xj}.h^{yj}$

Figure 1: New Signatures Derivation from existing signature and a derived signature

$$\alpha\ j,k = H(m2) + (xj - xk)$$

$$\beta\ j,k = H(m2') + (yj - yk)$$

$$\gamma\ j,k = m2' g^{H(m2)\cdot}h^{H(m2')}$$

Similar to (i, j), (j, k) is also modified. This signature is also published by Alice. If a signature is required related to both T1 and T2, Alice can sign using the unique pair of secret keys earlier used to generate signatures on m1,m1’ and m2,m2’, i.e. (xi, yi), (xk, yk). The signature will be (i, k, α i,k, β i,k, γ i,k,m1,m2,m1’,m2’) where

$$\alpha\ i,k = H(m1) + H(m2) + (xi - xk)$$

$$\beta\ i,k = H(m1') + H(m2') + (yi - yk)$$

$$\gamma\ i,k = (m1' + m2')\ g^{H(m1)+H(m2)} . h^{H(m1')+H(m2')}$$

We observe that for the message m1,m1’,m2,m2’, the above generated signature can be derived using the signature on m1,m1’ and m2,m2’. Let us call this derived signature as D1.

$$\alpha\ i,k = \alpha\ i,j + \alpha\ j,k$$

$$= H(m1) + (xi - xj) + H(m2) + (xj - xk)$$

$$= H(m1) + H(m2) + (xi - xk)$$

$$\beta\ i,k = \beta\ i,j + \beta\ j,k$$

$$= H(m1') + (yi - yj) + H(m2') + (yj - yk)$$

$$= H(m1') + H(m2') + (yi - yk)$$

$$\gamma\ i,k = \gamma\ i,j.\ \gamma\ j,k.(m1'.m2')^{-1}.(m1 + m2)$$

$$= m1'.g^{H(m1)\cdot}h^{H(m1')} .\ m2'.\ g^{H(m2)}.\ h^{H(m2')}. (m1'.m2')-1.(m1' + m2')$$

$$= (m1' + m2').g^{H(m1)+H(m2)} . h^{H(m1')+H(m2')}$$

If a third transaction T3 is required, a new node l can be created with secret keys (xl, yl) and public key vl = g^{xl} . h^{yl} . Let m3 be the data message sent by the customer and m3’ be the numerical value related to transaction T3. To sign the messages m3,m3’, Alice creates the signature, (k, l, αk,l, βk,l,γk,l,m3,m3’), where

$$\alpha k,l = H(m3) + (xk - xl)$$

$$\beta k,l = H(m3') + (yk - yl)$$

$$\gamma k,l = m3'\ g^{H(m3)} .h^{H(m3')}$$

To derive a signature on messages of transactions of T1, T2 and T3, we can use signature of D1 (as D1 signature is derived from signatures on messages on T1 and T2) and signature related related to T3, . Thus whenever messages of new transaction are to be signed, a new node can be created with secret keys and public key and attached to the previous transaction node. To obtain signature on all the messages till this new transaction, signature of the new transaction and the previous derived signature can be used.

DOI: 01.ijsip.01.01.02

Any signature can be verified using the following equation: Let us verify the signature
$(i, j, \alpha\ i,j\ , \beta\ i,j\ , \gamma\ i,j\ ,m1,m1')$ created for transaction T1.

$$vi.i,j = (m1' + m2').vj.g^{\alpha i,j} . h^{\beta i,j} \tag{2}$$

$$RHS = m1'.g^{xj} . h^{yj} .g^{H(m1)+(xi-xj)} . h^{H(m1') + (yi-yj)}$$

$$= m1'.g^{xj} . h^{yj} .g^{H(m1)}.g^{xi} .g^{-xj} .h^{H(m1')}.h^{yi} . h^{-yj}$$

$$= m1'.g^{H(m1)}.h^{H(m1')}.g^{xi}. h^{yi}$$

$$= \gamma\ i,j . vi$$

$$= LHS.$$

(m1'+m2') in the verification equation helps the verifier to substitute the values received from the signers and the verify the validity of the values. It must be noted that each customer transactions must be handled separately by creating a different set of nodes.

CONCLUSION

We have considered a scenario in banking environment where there is need to frequently issue signed receipts for the e-cheque deposited by the payee. Here, messages are signed as and when the related subprocess are completed. In our initial work on New Signature Derivation, we have come up with a method in which customers themselves can generate such signed receipts based on the signatures available on already completed transactions without the intervention of the bank which inturn reduces the work load on the bank. In all the signature derivations we make, we take care that a new signature derived is similar to the one that the signer would have generated if he had signed himself and also all signatures either existing or derived are verified using the same verification equation.

REFERENCES

[1] S. Micali, R.L. Rivest: *Transitive Signature Schemes*, CT-RSA 2002: 236- 243.

[2] M. Bellare and G. Neven. *Transitive Signatures based on Factoring and RSA*. Advances in Cryptology - Asiacrypt 2002 Proceedings, Lecture Notes in Computer Science Vol. 2501, Y. Zheng ed, Springer-Verlag, 2002.

[3] Damgard, I.: *Collision-free hash functions and public key signature schemes*. In: EUROCRYPT 87, LNCS, Vol.304, pp. 203216, Springer- Verlag, (1987).

[4] Taher ElGamal*: A Public Cryptosystem and a Signature Scheme based on Discrete Logarithms*, IEEE transactions on Information Theory, Vol. IT-31, No.4, (1985).

[5] FIPS 186. *Digital signature standard*. Federal Information Processing Standards Publication 186, U.S. Dept. of Commerce/NIST, National Technical Information Service, Springfield, Virginia, 1994.

[6] David J. Olkowski, Jr.,*Information Security Issues in E-Commerce* ,SANS GIAC Security Essentials March 26, 2001.

[7] Randy C. Marchany , Joseph G. Tront, *E-Commerce Security Issues*, Proceedings of the 35th Hawaii International Conference on System Sciences - 2002.

[8] Burt Kaliski : *RSA Digital Signature Standards*, RSA laboratories, 23rd National Information Systems Security Conference, Oct.16-19, 2000.

DOI: 01.ijsip.01.01.02